A conceptual framework to validate new features of corporate software, including client stakeholders

Flávio Henrique Alves
Computing Department
UTFPR - Federal University of Technology – Paraná
Cornélio Procópio, Brazil
fhalves@live.com
(000-0003-0806-9726)

Alexandre L'Erario
Computing Department
UTFPR - Federal University of Technology – Paraná
Cornélio Procópio, Brazil
alerario@utfpr.edu.br
(0000-0001-5233-7113)

José Antonio Gonçalves
Computing Department
UTFPR - Federal University of Technology – Paraná
Cornélio Procópio, Brazil
jgongalves@utfpr.edu.br

Abstract - This research for the practice of the full article presents the corporate systems - Enterprise Resource Planning (ERP) - that are necessary to meet the administrative demands of the sectors of an organization, such as production, finance, accounting, sales, purchasing, strategic monitoring, among others. **In order to serve the organization, an implanted ERP system can add new functionality and, in this evolutionary activity, validation can be complex and eminent, since a single business rule may require the cooperation of many stakeholders and their sectors. The validation of ERP's new features is necessary for maintaining their effectiveness. This work proposes a conceptual framework to allow professionals to validate a new business rule inserted in a functioning ERP system. To achieve this goal, it was in the main concepts of software engineering. The conceptual framework proposes to mitigate noise in the communication between the main actors in the deployment process - user and developer, resulting in delay, erroneous interpretations of requirements and indirect responsibilities, so that these factors influenced the deployment time, allowing for a significant impact improvement in future projects. The conceptual framework was composed from a case study in a Medium Organization. The ERP evolution process presents specific management challenges in relation to processes, people, products and technology.**

Keywords — conceptual framework, deployment ERP, software validation, stakeholders

I. INTRODUCTION

The Enterprise Resource Planning (ERP) software is characterize as a software package, considered as a technological strategy to improve, monitor or orchestrate the business performance of organizations. An ERP commonly brings together essential operations in the organization's business process such as production, finance, accounting, sales, purchasing, and strategic monitoring, among others. The implementation and implementation of this, requires extensive preparation, strategic planning, business processes, people management, brainstorming, architecture and software engineering.

However, an organization is dynamic and, therefore, new business demands arise frequently. The result of this scenario is the successive demand for new functional and non-functional requirements that need to be incorporate into the ERP in operation. Modifications from new deployments can have a major impact on the organization.

For a new functionality to be developed and successfully implemented, success in the validation phase is necessary, which is defined as "the process of determining the degree to which a model or simulation is an accurate representation of the real world, from the perspective of the intended uses. model or simulation "[1].

In a Medium / Large Enterprise [2], after the implementation of a new functionality, many ERP users need some time to familiarize themselves, to assimilate new concepts, technology and frequent use of them. After this phase, the system becomes reliable, robust and friendly to the organization, making the adopted ERP a natural and necessary routine for users, to fulfill their roles. The scope of this work is in the software validation process that takes place after implementation. Validation is the last stage of product delivery from the ERP customer's point of view. That is, after validation, the new functionality will be effectively available for use. However, the validation of a functionality in an ERP system is complex because it is common for a single business rule to require the participation of several departments and people in the organization or even people / systems outside it.

The objective of this work is to propose a conceptual framework whose goal is to empower IT professionals in the software validation process in the aforementioned environment.

In the Medium and / or Large Enterprise scenario, the validation of new features of an ERP can be too complex, for this reason, when using this conceptual framework, the IT professional helps in:

- Conduct strategies and direction in the execution of the validation of a new functionality;
- Focus mainly on functionality;
- Make a decision on the result of the validation;
- Manage the time for IT staff and users;
- Improve the reliability of the business rule from the customer's point of view.

The rest of this article is organized as follows. In Section 2, a background is presented. In Section 3, generation of the concept map. In Section 4, an overview of related work is presented. Finally, Section 5 concludes our work and suggests guidelines for future work.

II. THEORETICAL FOUNDATION

This section presents the theoretical basis used to provide a theoretical foundation of the main concepts used in this work.

A. ERP Systems

In the mid-1960s, corporations were in need of an automated business architecture system, and then the emergence of Materials Requirement Planning (MRP) was introduced to meet requirements that add to the final product. [3]. Not so late in the 90s, as it is known today, Enterprise Resource Planning (ERP) arises, proposing the control of the organization's resources. Making it a differential between

organizations, this fact led to a growing demand for ERP suppliers and their most diverse modules that could incorporate the product into the most complete product. [4].

ERP starts to gain value when team members incorporate tacit knowledge into it, this leads to the indispensable involvement of ERP in the processes [5]. Over the years, ERP has been on a journey of constant evolution and increasingly becoming an essential key for organizations. As the business environment accelerated, with the industrial revolution, formerly traditional businesses also added new technology to their e-business processes, so the MRP architecture also evolved into an automated and easily manageable process flow [6]. New technologies suggested and added features that made ERP flexible, for example with the Internet connection that made it more efficient.

Generally, ERP uses efforts to integrate information from Medium and Large Companies, since there are several challenges for a Large Company to control different processes to help control ERP becomes indispensable. Study conducted by [7] with the Saudi Aramco Company, which has 45,000 employees and produces 3.4 billion barrels and controls 100 um oil and gas fields within the KSA, which produce 253 trillion crude gas and 264 billion barrels of oil. It demonstrates that the ERP is a distributed System becoming indispensable for its activities.

The sector of distributors and manufacturers of automotive parts has been studied and according to the available data, more than 3,000 automotive companies worldwide rely on SAP software and automotive companies that are customers of SAP manufacture 77,000 vehicles every day. [8]. Each segment interconnects with several sub-systems from different sectors this generates a complex and necessary chain, since systems are alter for reasons inherent to the environment in which they are maintained, since they are interdependent.

B. Software Evolution

Software System development, maintenance and update process, are points that discriminate Software Evolution [9] necessary during the software life cycle, allowing adding or improving resources in the system. Otherwise, these factors influence the structure and process of a software, requiring that the internal and external quality attributes be modified continuously. Lehman [9] verified the evolution of large and consistent software and proposed seven laws, which are widely known and widespread as Lehman's Laws.

These laws, or rather, empirical hypotheses, induce that any software system used frequently must continually change to satisfy its stakeholders. [10].

Then the improvements must be, adapted and corrected to remain effective in constant evolution of the environment. Thus, the evolution of such systems is a complex event typified by feedback at various levels, various interactions and agents [11].

Over the years, authors have directed efforts in some investigative studies, aiming to validate Lehman's laws in software [12][13]. Due to extensive discussion of alternative research lines, Lehman's laws analyzed the evolution of software growth and characterized the evolution of its internal structure [14][15]. In fact, the evolution of software has yielded several studies, points remain open [16]. It is necessary to understand how software evolution occurs in a clearer way, taking into account its stakeholders.

As a software system evolves, its quality drops and its complexity increases, however, it is not known if there is a pattern for this degradation, and what factors lead to these conditions. Second Pan [17], improper sharing of information between the deployment team and members of the organization can result in an unsuccessful deployment. In situations where ERP implementation takes place in a geographically distributed environment, where different groups work together to carry out project tasks from different geographical locations [10] the distribution of knowledge is important to establish trust and improve the effectiveness of teamwork.

According to Camilo [10] who identified the structure of the software evolution process distributed in a large organization, he concluded that there are many stakeholders, such as internal developers, contractors for outsourced services, regulatory agency, government and so on.

For each new requirement, a risk / impact analysis is required, according to [10] for each stage of development; Reversals can occur and are not that simple; Each new deployment can have a considerable volume of users; This work did not address validation, but development and implementation.

C. Validation of ERP Systems

In the framework of ISO/IEC/IEEE - 12207-2017 [18], software validation has the principle that: a) the results to be achieved are defined; b) analyzes are conducted to assess the ability of design and development results to meet requirements; c) verification activities are carried out to ensure that the design and development outputs meet the input requirements. For such specifications it is necessary the involvement and lengths of all the actors, any noise can damage the postponement of the adherence of a new functionality to be implement.

If dealing with a new functionality, the impact on the correct processes may be irreversible, pausing or postponing processes for an off-line functionality integration may not be interesting, the possibility of changes being made on-the-fly [19], that is, without interference, it can be of great benefit.

The validation of ERP software in Medium / Large Organizations is an activity that can involve many users and variables, since a single business rule may require the contribution of several sectors of a company or external organizations.

In this sense, validating a new functionality in the ERP environment covered in this work can present problems such as:

a) Stakeholder time with communication

Scheduling and communicating to everyone involved in a business rule becomes a constant problem. The validation of the functionality can occur in production (on the fly), that is, no process can be interrupted for a long time or very often. In this sense, the rule is effectively performed by users and assisted in some way (monitored, accompanied).

By the IT, staff time is also a problem when this activity is simulated before going into production, that is, users can simulate a business process and check the quality of the implemented feature. However, in this case there is a waste of time executing a business rule that is not real.

Study conducted [20] evidenced problems in the validation of the implanted ERP, identifies that factors that depend on the proper functioning of the functionality, caused that even after 4 years of the implanted ERP there was customer dissatisfaction. This time spent negatively influenced organizational processes.

b) Organizational cultural change

For there to be impartiality it is necessary that everyone involved is willing to accept the changes in processes that a new functionality brings, the lack of support in the use of the functionality can influence and mask the proposed improvement or discovery of a problem related to functionality. For companies to continue to expand, assistance from stakeholders is needed to succeed in this new environment [21].

c) Changing organizational processes

Although ERP is a system that integrates information and for this reason, it changes or suggests that organizational processes be conduct within a standard or methodology that in most cases is different from the organization's processes. This difference directly affects production, since in addition to the new functionality the process in which the employee was used to performing has been changed.

To minimize these problems resulting from the evolution of ERP, the objective of this work is to develop a conceptual framework to enable IT (Information Technology) professionals during the validation process.

According [10] there are many involved in the organizational processes, which increases the complexity of the development process. Although requirements analysis and quality models minimize this final impact, it is common for misinterpretations and ambiguities to occur even in the phase of eliciting new requirements.

In this sense, validation seeks to consolidate development by making the new ERP functionality robust from the organization's point of view as soon as possible. There are many works focused on software testing such as [22], [23] but few focused on validation. It is important to emphasize that software validation involves two main parts: the developer and the client, and after this phase, the new functionality can be effectively employed.

III. CONCEPTUAL FRAMEWORK

A conceptual structure is a union of definitions of concepts, which can be used to describe and analyze phenomena, to ask research questions, to generalize states about phenomena, to specify models of mechanisms among other modes. For example, an object orientation theory could contain an explanation of polymorphisms and classes. Definition techniques are used to describe and analyze phenomena.

It should facilitate the distributed understanding of different, complementary meanings, giving value and assessing stakeholders. They include the operational part and execution aids, those who use the results of the assessment and all actors who provide perceptions of value or other information for the assessment.

Theories can consist of just a conceptual framework [24] defined a conceptual model of multiple component defects, containing definitions of concepts such as defect, multiple component defect, and architectural access point and repair dependency. Using this structure, phenomena that exhibit regularities can be described. For example, in the case investigated, a relationship was observed between maintenance cost, number of defects of multiple components and persistence of defects.

It is possible to verify that there is a difference between the generality of a conceptual structure and the generality of descriptions made using the conceptual structure. Since a conceptual framework is a set of definitions, it cannot be true or false, but it can be applicable or no. Generalization of a conceptual framework occurs when applied to many phenomena. On the other hand, descriptions can be true or false. A description is general if it is true for many phenomena. In the case studied by Li [24], 20% of the components contained 80% of the defects of multiple components. This is a true description of their case. It may well be false in other cases [25].

Often the material available in a deployment is direct to the IT area, such as requirements gathering and specification of the business rule. These details of shape are dammed with the development team and consultants, said technical part or part of the IT area [26].

I. Conceptual Framework Proposal

In fact, the involvement between IT and Administrative staff should happen, however it is unclear what the impacts are, little evidence and materials that can help both areas, visualize the impact of this interaction or the lack of it.

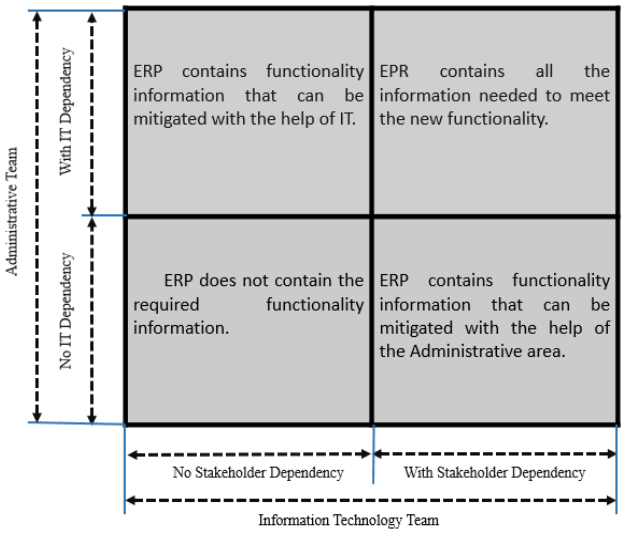

Fig. 1 Conceptual structure: shows the impact of the information exchange between the IT and Functional Areas, in the implementation of a new functionality.

Fig. 1 is a Conceptual Framework that helps in understanding the relationship between stakeholders, which implies the relationship of information exchange that each team has, so that the greater the involvement, the more complete the intended functionality becomes. In a situation of lack of communication from both parties, the functionality is exposed to not reaching the level of maturity necessary to complete the objective.

Considering the functionality to be validated, Fig. 1 represents the knowledge/information dependence of each one of the main involved. This chart indicates the degree of

dependence of functionality in relation to the IT team and the client (Administrative team).

In the same way, the greater the relationship of the parties, the better the maturity level of the functionality added to the ERP, which tends to absorb all the needs of the environment, being able to mitigate future problems. As, for example, the lack of information could lead to early maintenance, this impact is not mapped but induces rework, costs and wear and tear of those involved.

The workflow proposes a strong relationship between the stakeholders, the information must reach the most extreme ends of the process, in a homogeneous way, the breaking of this relationship influences the others causing uncontrolled.

The validation process can be composed for 5 activities, according to Fig. 2. Although previous ERP studies mention the importance of the implementation phase and the critical subsequent factors in the, studies did not provide a comprehensive recommendation for ERP management during the deployment phase of a new functionality. That is due to the focus of the previous study concentrating efforts on a broad aspect for deployment of a complete ERP and not for adding a functionality [27].

In this scenario, we see the need for off-line validation assuming the mitigation of failures in the go-live phase and customer dissatisfaction. Fig. 1 was elaborated from the current literature and the case study in the client company of the service sector, integrating existing processes with non-existent ones detected the lack of empowering of the TI professional and absence of steps that seems essential for the success of the implementation of a new functionality in EPR.

The phases of Off-line Validation Processes of a new functionality in proprietary software in the literature seem to be absent [28][29].

The validation process is placed within the deployment activity. Deployment in our case is the process whose final result is the availability of the new version of the software to the end customer.

In this case, a deployment process for an ERP, in the scenario described, can be configured in 5 activities. Such activities include two types of validation, the first of which can be performed before implementation (off-line validation) and the second after implementation (on the fly). These are not mutually exclusive and may, at the discretion of the IT organization and the customer, execute only one, neither, or both.

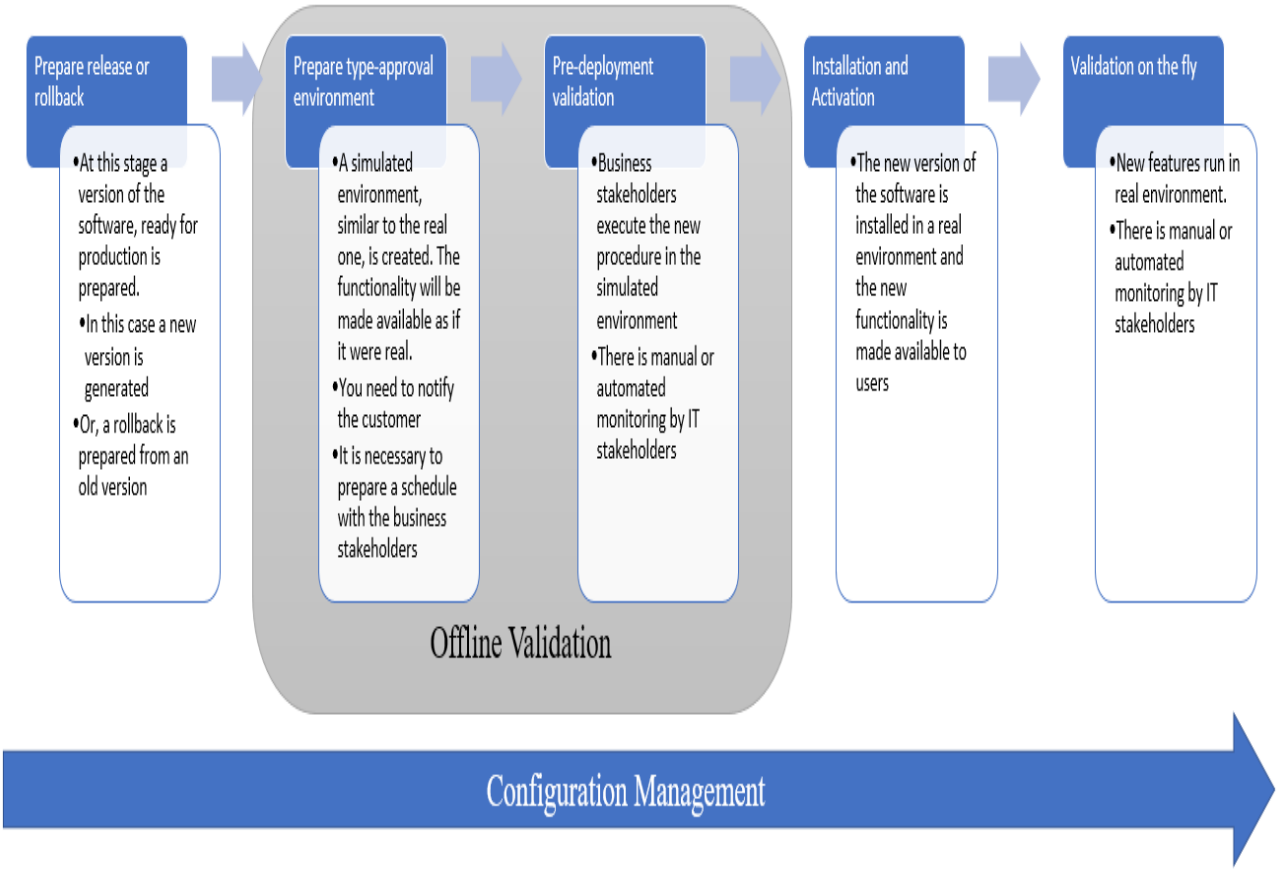

Fig. 2. Validation Process (Deploy).

The empower of IT professionals in the processes, combined with the conceptual structure (Fig. 1) promotes better interaction between IT and customers and supports the validation of the product, making it reach acceptable levels of quality.

Table I elicits in form system requirements the phases of the Validation Process - Deploy, this information served as a basis for the relationship between validation of activities and conceptual model, in addition to helping to identify the level by activity, dependency vs. scenario vs. relevance.

TABLE I. RELATIONSHIP BETWEEN VALIDATION ACTIVITIES AND THE CONCEPTUAL FRAMEWORK.

ID	Objective	Activities
1	Prepare release or rollback	1.1 At this stage a version of the software, ready for production is prepared. In this case a new version is generated or, a rollback is prepared from an old version
2	Prepare type-approval environment	2.1 A simulated environment, similar to the real one, is created. The feature will be made available as if it were real. 2.2 It is necessary to notify the client 2.3 It is necessary to prepare a schedule with the business stakeholders
3	Pre-deployment validation	3.1 The stakeh. execute the new procedure in the simulated environment 3.2 There is a manual or automated monitoring by IT stakeholders
4	Installation and Activation	4.1 The new version of the software is installed in a real environment and the new functionality is made available to users
5	Validation on the fly	5.1 New features run in a real environment. 5.2 There is manual or automated monitoring by IT stakeholders

It is necessary to evaluate each scenario based on the activity to visualize the dependency of each actor and what are his or her impacts on the final product. As all activity has its role in the validation process, relevance will be treat as high.

TABLE II. LEVEL IDENTIFICATION BY ACTIVITY - DEPENDENCY VS. SCENARIO VS. RELEVANCE.

Dependency - IT Departmen vs Stakeholders	NO | NO				YES | NO				NO | YES				YES | YES			
Activity	Relevance(?)															
	Communication				Culture				Process				Communication			
1.1	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
2.1	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
2.2	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
2.3	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
3.1	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
3.2	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
4.1	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
5.1	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
5.2	L	A	A	H	L	A	A	H	L	A	A	H	L	A	A	H
	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 1	Scenario 2	Scenario 3	Scenario 4

Subtitle:

NO | NO - No dependency on IT Department vs No dependence on Stakeholders.

YES | NO - Dependent on IT Department vs No dependence on Stakeholders.

NO | YES - No dependency on IT department vs With dependence on Stakeholders.

YES | YES - Dependent on IT Department vs With dependence on Stakeholders.

L – Low | A – Medium | H - High

Table II indicates 4 possible scenarios identified in the case study are these: Scenario 1, indicated by Table II - ERP does not contain the necessary functionality information. This scenario shows the processes running without the relationship between those involved, this when tasks are performed separately, this causes the success rate to be low, due to the distance between the IT areas and customer stakeholders.

In Scenario 2 - The ERP contains information on the functionality that can be mitigated with the help of IT. At the moment when there is an effort by IT to receive information, analyze the culture of the organization and stick to the reflex that other processes of the organization reach, the success rate is average.

Scenario 3 - ERP contains information about the functionality that can be explored with the assistance of the Administrative. When there is an effort by the administrative area, it provides information with the support of the organization's culture, which maps and shows possible impacts on the change of processes in order to mitigate errors, the success rate is average.

Scenario 4 - The ERP contains all the information needed to meet the new functionality. With synchronous communication, those involved are aware of the importance of the new functionality, that is, culturally aware and that mapping all possible effects on changing organizational processes, it is possible to achieve a high rate in the implementation of a new functionality when validating itself. This scenario is ideal for all interested parties to be successful in validating the new functionality.

IV. METHODS AND PROCEDURES

This research is a case study in the area of software engineering due to an immense variety in processes, organizational culture and products; it was necessary to analyze context-specific factors in order to understand in depth the phenomena identified [30].

The study was carried out Client Company with the configuration of the scenario for implementation of a new functionality in the module of Materials and Purchasing Process - "Wizard Screen of Purchasing Process".

The client company is a company founded in 1992, participates in the service outsourcing business in the most diverse areas; among them are Reading and delivery of Energy, Water and Gas accounts. It has about 2300 employees, of which it has an administrative staff of 150 employees, all of whom depend on ERP functionalities, in the various segments such as production, finance, accounting, sales, purchasing, and strategic monitoring, among others.

The entire implementation process took place on the fly, the new functionality included resources from the purchasing department, and this functionality is related to several processes such as finance, accounting, sales, and inventory, among others. Many factors were neglected, this caused an impact on the correct processes, as the stakeholders were not aware of the changes, and soon it affected the work routine causing discontent.

V. RESULTS

After analyzing the organization, the following processes were found that:

Analyzing the nonexistent requirements of activities 1.1, 2.1, 2.2, 2.3, 3.1, 3.2 and 5.2, as shown in Table VI, it was possible to identify a low rate scenario in which the interested parties act independently, thus fitting into Scenario 1,

contained in Table II. In which it demonstrates the lack of relationship, which influenced a result of failure, among other factors, it is possible to see that the time of communication, the culture of organizational change and the change of processes, were trivial factors in the implementation of functionality.

TABLE III. VALIDATION OF NEW FUNCTIONALITY OF THE CLIENT COMPANY SUPPORTS THE PROPOSAL OF THE CONCEPTUAL FRAMEWORK MODEL.

Activity	Purchasing Process Wizard Screen	Consequence	Cause	Relevance		
				Communication	Culture	Processes
1.1	Nonexistent	The customer had no familiarity, there was no rollback	IT understands that it´s not necessary	Low	Low	Low
2.1	Nonexistent	Time was spent for presentation	IT understands that it´s not necessary	Low	Low	Low
2.2	Nonexistent	Notification was made late	IT notified the wrong department	Low	Low	Low
2.3	Nonexistent	Stakeholders were triggered as a problem appeared	IT called only the buyer to participate	Low	Low	Low
3.1	Nonexistent	No offline environment	IT understands that it´s not necessary	Low	Low	Low
3.2	Nonexistent	No offline environment	IT understands that it´s not necessary	Low	Low	Low
4.1	Existing	The employees did not know the new functionality	TI late notified the new functionality	Low	Low	Low
5.1	Existing	Many incocistencia appeared to other departments	There was no planned preparat	Average	Average	Average
5.2	Nonexistent	Time spent on error notifications	IT finds it unnecessary	Low	Low	Low

Despite the existence of activities 4.1 and 5.2, the average and low rates respectively, since there is no high interaction between those involved the IT area and the administrative area. The lack of an off-line step was a major factor, without these steps many risks are no longer mitigated, leading to the validation of the unfeasible implementation.

VI. EMPOWER OF PROFESSIONALS OF IT

In his article Camilo [10] through literature, interviews and documentation, identifies stakeholders. Thus helping to empower the IT professional in this identification, since it is known that each project has its particularity, the ones involved are variable, the scenario found changes over time and the human material can influence the evolution of the project.

TABLE IV. SIMPLIFIED GUIDE TO IDENTIFYING DEPENDENCIES AND RELEVANCE

Dependency		Relevance(?)		
IT Department	Stakeholders	Communication	Culture	Process
No	No	Low	Low	Low
Yes	No	Average	Average	Average
No	Yes	Average	Average	Average
Yes	Yes	High	High	High

Given Table IV, the following step helps management:

- Empower the employee to identify the scenarios (IT dependence and stakeholders): The employee can investigate the situation of the dependencies and treat each relevance in isolation until he/she obtains the support of both parties. Similarly, it is possible to identify that certain relevance does not reach the expected and then investigate which areas are not corroborating for the evolution of the objective, thus attacking the problem area punctually.

- Empower the employee to identify stakeholders and pay attention to communication: Once a low or medium rate in communication is identified, either through e-mail, interviews, informal conversation or clear level of lack of knowledge on the part of the client; - it is not trivial to make efforts to create a bond and strengthen communication, to resolve such a meeting, email or mailing list through mobile digital media with those involved can result in good results.

- Empower the employee to adapt the validation process: customize Fig. 4 according to demand. Understand the cost of each activity and the consequence of not including it in the process.

- Empower the employee to prepare a checklist for validation, regardless of the requirement, but supported by the demands indicated by the IT and stakeholders dependencies: Achieving the involvement of both parties is the objective of having a good level of the project, once a low level has been identified or medium fee or absence of dependency. The correction must be immediate and remedied one by one until high dependence and involvement of both parties are reached, thus ensuring that the new ERP functionality contains all the information necessary to meet its purpose regardless of the project.

- Empower the employee to follow the process and provide priority information on non-conformities in the execution of the validation: When identifying failures or lack of commitment, this information must be part of a report reported to senior management, who must agree with the necessary changes, already identified and scored. However, nothing is valid if there is no support, neglecting the data collected and the proposed action plan may compromise the success of the project or less minimize the potential that can be reached.

VII. CONCLUSIONS

The absence of processes 1.1, 2.1, 2.2, 2.3, 3.1, 3.2 and 5.2 caused problems in the time of interested parties with communication, organizational cultural change and alteration of organizational processes in this way, we corroborate when mentioning that the processes covered in this article are essential for software validation in this scenario.

With this study of the current literature and the case study in the client company, it was possible to elaborate the conceptual framework to validate new features of the corporate software, identifying that communication, organizational culture and processes are preponderant factors for success and good development and if the objective could be achieved, the elicitation of the activity in Table I gave the possibility of verifying the processes, which combined with Table II allows validation of a corporate scenario, so that the interactions benefit the processes and extend to the client stakeholders.

The work proposal establishes a structured set of activities and techniques that contribute and guide the requirements of developers, engineers, requirements analysts and others, to obtain a complete and objective specification, analysis and evaluation of the requirements and iteratively refines them if necessary the stakeholders . The IT employee needs five empowering (scenarios, stakeholders, process, checklist, execution). In this way, the expected results will be achieved ensuring that the requirements effectively generated the agreed objective.

With these skills, the team can regulate the best validation process according to the characteristics of those involved. It becomes feasible to equalize the cost of including or not including a certain activity within the deployment process. In this way, the time demand of the necessary teams is also regulated, since the first action to be developed is to map the requirement in one of the four scenarios, this strategy will assist the execution of the validation of a new functionality. With the validation completed, it is up to the professional to make decisions in order to eliminate gaps found, which must be presented to the client and the interested parties, seeking to obtain timesaving in the development and better understanding of the business rules.

As future work, the other scenarios will be verified in other case studies. Problems can occur alone or with a specific actor, it is necessary to inspect and evaluate the impact of these factors, which can unite more than one scenario in the validation of the same functionality. New factors that threaten the success of software deployment [10] can be added, such as employee turnover, data standards, distributed software, among others.

Finally, we can highlight that the main contribution of this work is a tool that helps in the validation of a new functionality in the evolution of the software through a context that simulates a situation that occurs off-line and, later, in motion, in a context of great importance that is not commonly presented to IT professionals. This tool should also help client companies and stakeholders companies.

REFERENCES

[1] D. E. Stevenson, "Verification and validation of complex systems," *Intell. Eng. Syst. through Artif. Neural Networks*, pp. 159–164, 2002.

[2] C. Y. Laporte, N. Séguin, G. Villas Boas, and S. Buasung, "Small tech firms: Seizing the benefits of software and systems engineering standards," *ISO Focus.*, vol. 4, no. 2, pp. 32–36, 2013.

[3] M. A. Rashid, L. Hossain, and J. D. Patrick, "The evolution of ERP systems: A historical perspective," in *Enterprise Resource Planning: Solutions and Management*, IGI global, 2002, pp. 35–50.

[4] E. Callaway, *ERP-the next generation: ERP is web enabled for e-business*. Computer Technology Research Corp., 2000.

[5] Z. Lee and J. Lee, "An ERP implementation case study from a knowledge transfer perspective," *J. Inf. Technol.*, vol. 15, no. 4, pp. 281–288, 2000.

[6] R. Baskerville, S. Pawlowski, and E. McLean, "Enterprise resource planning and organizational knowledge: patterns of convergence and divergence," *ICIS 2000 Proc.*, p. 36, 2000.

[7] A. Bajahzar, A. Alqahtani, and A. Baslem, "A survey study of the enterprise resource planning system," in *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, 2012, pp. 246–252.

[8] A. Lorenc and M. Szkoda, "Customer logistic service in the automotive industry with the use of the SAP ERP system," in *2015 4th International conference on advanced logistics and transport (ICALT)*, 2015, pp. 18–23.

[9] M. M. Lehman, "Leis da evolução do software revisitadas," in *Workshop Europeu sobre Tecnologia de Processos de Software*, pp. 108–124.

[10] J. R. M. Camilo, A. L'Erario, T. Pagotto, and J. A. Fabri, "A process for distributed software evolution: A proprietary software case study," in *Proceedings - International Conference on Software Engineering*, 2018, pp. 44–53.

[11] T. Mens, Y.-G. Guéhéneuc, J. Fernandez-Ramil, and M. D'Hondt, "Guest Editors' Introduction: Software Evolution," *IEEE Softw.*, vol. 27, pp. 22–25, 2010.

[12] R. P. de Oliveira, A. R. Santos, E. S. de Almeida, and G. S. da Silva Gomes, "Evaluating Lehman's Laws of software evolution within software product lines industrial projects," *J. Syst. Softw.*, vol. 131, pp. 347–365, 2017.

[13] G. Xie, J. Chen, and I. Neamtiu, "Towards a better understanding of software evolution: An empirical study on open source software," in *2009 IEEE International Conference on Software Maintenance*, 2009, pp. 51–60.

[14] S. Koch, "Software evolution in open source projects—a large-scale investigation," *J. Softw. Maint. Evol. Res. Pract.*, vol. 19, no. 6, pp. 361–382, 2007.

[15] R. P. F. Trindade, T. S. Orfanó, K. A. M. Ferreira, and E. F. Wanner, "The dance of classes-A stochastic model for software structure evolution," in *2017 IEEE/ACM 8th Workshop on Emerging Trends in Software Metrics (WETSoM)*, 2017, pp. 22–28.

[16] T. Mens, M. Wermelinger, S. Ducasse, S. Demeyer, R. Hirschfeld, and M. Jazayeri, "Challenges in software evolution," in *Eighth International Workshop on Principles of Software Evolution (IWPSE'05)*, 2005, pp. 13–22.

[17] S. Pan, S. Newell, J. Huang, and A. Cheung, "Knowledge integration as a key problem in an ERP implementation," *ICIS 2001 Proc.*, p. 36, 2001.

[18] "ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes - Redline," *ISO/IEC/IEEE 122072017(E) First Ed. 2017-11 - Redline*, pp. 1–462, 2017.

[19] M. Happe, F. Meyer auf der Heide, P. Kling, M. Platzner, and C. Plessl, "On-The-Fly Computing: A novel paradigm for individualized IT services," in *16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013)*, 2013, pp. 1–10.

[20] K. Shi and Q. Lu, "Exploration and verification of factors in the Front-end stage of the ERP Implementation Process: Evidence from case study and survey research," in *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 2009, pp. 1538–1542.

[21] J. M. Hussey and S. E. Hall, *Managing global development risk*. CRC Press, 2007.

[22] S. Wieczorek, A. Stefanescu, and I. Schieferdecker, "Test Data Provision for ERP Systems," in *2008 1st International Conference on Software Testing, Verification, and Validation*, 2008, pp. 396–403.

[23] S. Bekrar, C. Bekrar, R. Groz, and L. Mounier, "Finding Software Vulnerabilities by Smart

Fuzzing," in *2011 Fourth IEEE International Conference on Software Testing, Verification and Validation*, 2011, pp. 427–430.

[24] Z. Li *et al.*, "Characteristics of multiple-component defects and architectural hotspots: A large system case study," *Empir. Softw. Eng.*, vol. 16, no. 5, pp. 667–702, 2011.

[25] R. Wieringa and M. Daneva, "Six strategies for generalizing software engineering theories," *Sci. Comput. Program.*, vol. 101, pp. 136–152, 2015.

[26] P. Bingi, M. K. Sharma, and J. K. Godla, "Critical issues affecting an ERP implementation," *IS Manag.*, vol. 16, no. 3, pp. 7–14, 1999.

[27] D. R. Chandra and R. Govindaraju, "Recommendation of ERP post-project management," in *2012 IEEE International Conference on Management of Innovation & Technology (ICMIT)*, 2012, pp. 418–423.

[28] G. Salvendy, *Handbook of industrial engineering: technology and operations management*. John Wiley & Sons, 2001.

[29] G. Shanks, P. B. Seddon, and L. P. Willcocks, *Second-wave enterprise resource planning systems: Implementing for effectiveness*. Cambridge University Press, 2003.

[30] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.